Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1462	(714/38).CCLS.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/14 13:10
L2	38608	(stall$3 or lock$4 or locked-up or crash$4 or dead$4) with (thread or routine)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/14 13:10
L3	57	L1 and L2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/14 13:10
S1	35797	(stall$3 or lock$4 or locked-up or crash$4 or dead$4) with (thread or routine)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/14 12:09
S2	1614	(software or program) same ((stall$3 or lock$4 or locked-up or crash$4 or dead$4) with (thread or routine))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:10
S3	9216	object adj code	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:10
S4	64	((software or program) same ((stall$3 or lock$4 or locked-up or crash$4 or dead$4) with (thread or routine))) and (object adj code)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/14 12:08
S5	613	(stall$3) with (thread or routine)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:28
S6	14	(object adj code) and ((stall$3) with (thread or routine))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:21

S7	155	((stall$3) with (thread or routine)) same (software or program)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:21
S8	112	(timer or counter) and (((stall$3) with (thread or routine)) same (software or program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:22
S9	49339	timeout or (tim$3 adj out)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:22
S10	17	((timer or counter) and (((stall$3) with (thread or routine)) same (software or program))) and (timeout or (tim$3 adj out))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:22
S11	0	(stall$3) with (thread or routine)with symmantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:26
S12	0	(stall$3) with (thread or routine)with symantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:24
S13	0	(stall$3) with (thread or routine) and symantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:24
S14	0	(stall$3) with (thread or routine) and symmantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:24
S15	5	symmantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:24
S16	811	symantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:24

S17	5	symantec and wind	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:25
S18	563	symantec and (fault or error or problem) and software	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:25
S19	4	symantec and (fault or error or problem) and software and wind	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:26
S20	72	norton and (fault or error or problem) and software and wind	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:26
S21	0	((stall$3) with (thread or routine)) and norton	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:27
S22	0	((stall$3) with (thread or routine)) and symmantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:27
S23	0	((stall$3) with (thread or routine)) and symantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:27
S24	0	unwond and symantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:27
S25	3	unwind and symantec	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:28
S26	979	(froz$4 or freez$4) with (thread or routine)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:28

S27	611	((froz$4 or freez$4) with (thread or routine)) and (fault or error or problem)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:29
S28	0	((froz$4 or freez$4) with (thread or routine)) and (fault or error or problem) and (timeout or (tim$3 adj out)) and ((timer or counter) and (((stall$3) with (thread or routine)) same (software or program)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:29
S29	101	((froz$4 or freez$4) with (thread or routine)) and (fault or error or problem) and (timeout or (tim$3 adj out))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:30
S30	101	((froz$4 or freez$4) with (thread or routine)) and (fault or error or problem) and (timeout or (tim$3 adj out)) and (program or software)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:30
S31	80	((froz$4 or freez$4) with (thread or routine)) and (fault or error or problem) and (timeout or (tim$3 adj out)) and (software)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:30
S32	80	(((froz$4 or freez$4) with (thread or routine)) and (fault or error or problem) and (timeout or (tim$3 adj out)) and (software)) not refridgerator	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:31
S33	73	(((froz$4 or freez$4) with (thread or routine)) and (fault or error or problem) and (timeout or (tim$3 adj out)) and (software)) not refrigerator	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/30 18:36
S34	1413	(714/38,35).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/08/30 18:37
S35	3715	((stall$4 or hung$4 or crash$4) with (thread or process or method or routine)) and (program)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/09/01 18:08

S36	28	(714/706).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/09/01 17:49
S37	841	timer and (((stall$4 or hung$4 or crash$4) with (thread or process or method or routine)) and (program))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/09/01 18:08
S38	222	timer and (((stall$4 or hung$4 or crash$4) with (thread or process or method or routine)) and (program)) and thread	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/09/01 18:08
S39	2	10/079966	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/09/05 10:36
S40	504959	(expect$3 or estimat$4 or extrapol$5 or due) with (time or period)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:48
S41	28034	(respon$5 or return$4) with S40	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:49
S42	659318	(stall$4 or freez$4 or frozen or hung or hang$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:57
S43	10646200	application or thread or method or routine or code or program	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:50
S44	74912	S42 with S43	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:51
S45	27	S41 same S44	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:56

S46	116591	overdue or "over due" or late	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:57
S47	9548	S46 with S43	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:57
S48	695870	(stall$4 or freez$4 or frozen or hung or hang$4 crash$)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:59
S49	1834	S47 and S48	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 10:59
S50	88	S47 same S48	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 11:35
S51	0	714/51,55	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/02 11:36
S52	410	(714/51,55).CCLS.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/02 11:36
S53	8	((software or program) same ((stall$3 or lock$4 or locked-up or crash$4 or dead$4) with (thread or routine))) and (object adj code) and @pd>"20050402"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/14 12:09
S54	1462	(714/38).CCLS.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/14 12:09
S55	38608	(stall$3 or lock$4 or locked-up or crash$4 or dead$4) with (thread or routine)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/07/14 12:09

Terms used **stalled routine instrumentation** Found **3,783** of **157,873**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Scalable analysis techniques for microprocessor performance counter metrics

Dong H. Ahn, Jeffrey S. Vetter

November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing**

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, citings, index terms

Contemporary microprocessors provide a rich set of integrated performance counters that allow application developers and system architects alike the opportunity to gather important information about workload behaviors. Current techniques for analyzing data produced from these counters use raw counts, ratios, and visualization techniques help users make decisions about their application performance. While these techniques are appropriate for analyzing data from one process, they do not scale easi ...

2 Continuous profiling: where have all the cycles gone?

Jennifer M. Anderson, Lance M. Berc, Jeffrey Dean, Sanjay Ghemawat, Monika R. Henzinger, Shun-Tak A. Leung, Richard L. Sites, Mark T. Vandevoorde, Carl A. Waldspurger, William E. Weihl

November 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 4

Full text available: pdf(259.35 KB) Additional Information: full citation, abstract, references, citings, index terms

This article describes the Digital Continuous Profiling Infrastructure, a sampling-based profiling system designed to run continuously on production systems. The system supports multiprocessors, works on unmodified executables, and collects profiles for entire systems, including user programs, shared libraries, and the operating system kernel. Samples are collected at a high rate (over 5200 samples/sec. per 333MHz processor), yet with low overhead (1–3% slowdown for most workloads). A ...

Keywords: performance understanding, performance-monitoring hardware, profiling, program analysis

3 Complexity/performance tradeoffs with non-blocking loads

K. I. Farkas, N. P. Jouppi

April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture**, Volume 22 Issue 2

Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

Non-blocking loads are a very effective technique for tolerating the cache-miss latency on data cache references. In this paper, we describe several methods for implementing non-blocking loads. A range of resulting hardware complexity/performance tradeoffs are investigated using an object-code translation and instrumentation system. We have investigated the SPEC92 benchmarks and have found that for the integer benchmarks, a simple hit-under-miss implementation achieves almost all of the availabl ...

4 Shasta: a low overhead, software-only approach for supporting fine-grain shared memory

Daniel J. Scales, Kourosh Gharachorloo, Chandramohan A. Thekkath

October 1996 **Proceedings of the seventh international conference on Architectural support for programming languages and operating systems**, Volume 30 , 31 Issue 5 , 9

Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes Shasta, a system that supports a shared address space in software on clusters of computers with physically distributed memory. A unique aspect of Shasta compared to most other software distributed shared memory systems is that shared data can be kept coherent at a fine granularity. In addition, the system allows the coherence granularity to vary across different shared data structures in a single application. Shasta implements the shared address space by transparently rewrit ...

5 Continuous profiling: where have all the cycles gone?

Jennifer M. Anderson, Lance M. Berc, Jeffrey Dean, Sanjay Ghemawat, Monika R. Henzinger, Shun-Tak A. Leung, Richard L. Sites, Mark T. Vandevoorde, Carl A. Waldspurger, William E. Weihl

October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles**, Volume 31 Issue 5

Full text available: pdf(2.29 MB) Additional Information: full citation, references, citings, index terms

6 Non-intrusive and interactive profiling in parasight

Ziya Aral, Ilya Gertner

January 1988 **ACM SIGPLAN Notices , Proceedings of the ACM/SIGPLAN conference on Parallel programming: experience with applications, languages and systems**, Volume 23 Issue 9

Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings, index terms

Debugging the performance of parallel applications is crucial for fully utilizing the potential of multiprocessor hardware. This paper describes profiling tools in Parasight, a programming environment that is geared towards non-intrusive performance analysis and high-level debugging. In Parasight, profilers execute as observer programs which run concurrently with the target program and monitor its behavior. This design grew out of our experience in debugging and monitoring the performance o ...

7 Exploiting hardware performance counters with flow and context sensitive profiling

Glenn Ammons, Thomas Ball, James R. Larus

May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation**, Volume 32 Issue 5

Full text available: pdf(1.67 MB) Additional Information: full citation, abstract, references, citings, index terms

A program profile attributes run-time costs to portions of a program's execution. Most profiling systems suffer from two major deficiencies: first, they only apportion simple

metrics, such as execution frequency or elapsed time to static, syntactic units, such as procedures or statements; second, they aggressively reduce the volume of information collected and reported, although aggregation can hide striking differences in program behavior.This paper addresses both concerns by exploiting the har ...

8 Computation: Understanding the efficiency of GPU algorithms for matrix-matrix multiplication

K. Fatahalian, J. Sugerman, P. Hanrahan

August 2004 **Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware**

Full text available: pdf(145.04 KB) Additional Information: full citation, abstract, references, index terms

Utilizing graphics hardware for general purpose numerical computations has become a topic of considerable interest. The implementation of streaming algorithms, typified by highly parallel computations with little reuse of input data, has been widely explored on GPUs. We relax the streaming model's constraint on input reuse and perform an in-depth analysis of dense matrix-matrix multiplication, which reuses each element of input matrices O(n) times. Its regular data access pattern and highly para ...

9 Experience with a software-defined machine architecture

David W. Wall

May 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 14 Issue 3

Full text available: pdf(2.86 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We have built a system in which the compiler back end and the linker work together to present an abstract machine at a considerably higher level than the actual machine. The intermediate language translated by the back end is the target language of all high-level compilers and is also the only assembly language generally available. This lets us do intermodule register allocation, which would be harder if some of the code in the program had come from a traditional assembler, out of sight of ...

Keywords: RISC, graph coloring, intermediate language, interprocedural, optimization, pipeline scheduling, profiling, register allocation, register windows

10 ATOM: a system for building customized program analysis tools

Amitabh Srivastava, Alan Eustace

June 1994 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation**, Volume 29 Issue 6

Full text available: pdf(965.20 KB) Additional Information: full citation, abstract, references, citings, index terms

ATOM (Analysis Tools with OM) is a single framework for building a wide range of customized program analysis tools. It provides the common infrastructure present in all code-instrumenting tools; this is the difficult and time-consuming part. The user simply defines the tool-specific details in instrumentation and analysis routines. Building a basic block counting tool like Pixie with ATOM requires only a page of code. ATOM, using OM link-time technology, organizes the final execu ...

11 Physical Experimentation with Prefetching Helper Threads on Intel's Hyper-Threaded Processors

Dongkeun Kim, Steve Shih-wei Liao, Perry H. Wang, Juan del Cuvillo, Xinmin Tian, Xiang Zou, Hong Wang, Donald Yeung, Milind Girkar, John P. Shen

March 2004 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Full text available: pdf(264.47 KB) Additional Information: full citation, abstract, citings, index terms

Pre-execution techniques have received much attention as aneffective way of prefetching cache blocks to tolerate the ever-increasingmemory latency. A number of pre-execution techniquesbased on hardware, compiler, or both have been proposed andstudied extensively by researchers. They report promising resultson simulators that model a Simultaneous Multithreading (SMT)processor. In this paper, we apply the helper threading idea ona real multithreaded machine, i.e., Intel Pentium 4 processor withHyp ...

12 Memory-wall: Profile-guided post-link stride prefetching

Chi-Keung Luk, Robert Muth, Harish Patil, Richard Weiss, P. Geoffrey Lowney, Robert Cohn
June 2002 **Proceedings of the 16th international conference on Supercomputing**

Full text available: pdf(281.59 KB) Additional Information: full citation, abstract, references, citings, index terms

Data prefetching is an effective approach to addressing the memory latency problem. While a few processors have implemented hardware-based data prefetching, the majority of modern processors support data-prefetch instructions and rely on compilers to automatically insert prefetches. However, most prefetching schemes in commercial compilers suffer from two limitations: (1) the source code must be available before prefetching can be applied, and (2) these prefetching schemes target only loops with ...

Keywords: address strides, data prefetching, memory latency, post-link optimizations, profiling

13 Efficient path profiling

Thomas Ball, James R. Larus
December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.29 MB) Additional Information: full citation, abstract, references, citings, index terms

A path profile determines how many times each acyclic path in a routine executes. This type of profiling subsumes the more common basic block and edge profiling, which only approximate path frequencies. Path profiles have many potential uses in program performance tuning, profile-directed compilation, and software test coverage. This paper describes a new algorithm for path profiling. This simple, fast algorithm selects and places profile instrumentation to minimize run-time overhead. Instrument ...

14 A general framework for prefetch scheduling in linked data structures and its application to multi-chain prefetching

Seungryul Choi, Nicholas Kohout, Sumit Pamnani, Dongkeun Kim, Donald Yeung
May 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 2

Full text available: pdf(2.45 MB) Additional Information: full citation, abstract, references, index terms

Pointer-chasing applications tend to traverse composite data structures consisting of multiple independent pointer chains. While the traversal of any single pointer chain leads to the serialization of memory operations, the traversal of independent pointer chains provides a source of memory parallelism. This article investigates exploiting such *interchain memory parallelism* for the purpose of memory latency tolerance, using a technique called *multi--chain prefetching*. Previous work ...

Keywords: Data prefetching, memory parallelism, pointer-chasing code

15 Ispike: A Post-link Optimizer for the Intel®Itanium®Architecture

Chi-Keung Luk, Robert Muth, Harish Patil, Robert Cohn, Geoff Lowney
March 2004 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**
Full text available: pdf(286.96 KB) Additional Information: full citation, abstract, citings, index terms

Ispike is post-link optimizer developed for theIntel®Itanium Processor Family (IPF) processors.TheIPF architecture poses both opportunities and challenges topost-link optimizations.IPF offers a rich set of performancecounters to collect detailed profile information at a low cost,which is essential to post-link optimization being practical.At the same time, the prediction and bundling features onIPF make post-link code transformation more challengingthan on other architectures.In Ispike, we have ...

16 Improving the performance of speculatively parallel applications on the Hydra CMP
Kunle Olukotun, Lance Hammond, Mark Willey
May 1999 **Proceedings of the 13th international conference on Supercomputing**
Full text available: pdf(1.66 MB) Additional Information: full citation, references, citings, index terms

Keywords: chip multiprocessor, data speculation, feedback-driven optimization, multithreading, parallel programming, performance evaluation

17 A Hardware-Software Platform for Intrusion Prevention
Milenko Drinic, Darko Kirovski
December 2004 **Proceedings of the 37th annual International Symposium on Microarchitecture**
Full text available: pdf(254.63 KB) Additional Information: full citation, abstract

Preventing execution of unauthorized software on a given computer plays a pivotal role in system security. The key problem is that although a program at the beginning of its execution can be verified as authentic, its execution flow can be redirected to externally injected malicious code using, for example, a buffer overflow exploit. We introduce a novel, simplified, hardware-assisted intrusion prevention platform. Our platform introduces overlapping of program execution and MAC verification. It ...

18 MemSpy: analyzing memory system bottlenecks in programs
Margaret Martonosi, Anoop Gupta, Thomas Anderson
June 1992 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1992 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems**, Volume 20 Issue 1
Full text available: pdf(1.57 MB) Additional Information: full citation, abstract, references, citings, index terms

To cope with the increasing difference between processor and main memory speeds, modern computer systems use deep memory hierarchies. In the presence of such hierarchies, the performance attained by an application is largely determined by its memory reference behavior—if most references hit in the cache, the performance is significantly higher than if most references have to go to main memory. Frequently, it is possible for the programmer to restructure the data or code to achieve bet ...

19 Embra: fast and flexible machine simulation
Emmett Witchel, Mendel Rosenblum
May 1996 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1996 ACM SIGMETRICS international conference on Measurement and modeling of computer systems**, Volume 24 Issue 1
Additional Information: full citation, abstract, references, citings, index

Full text available: pdf(1.83 MB) terms

This paper describes Embra, a simulator for the processors, caches, and memory systems of uniprocessors and cache-coherent multiprocessors. When running as part of the SimOS simulation environment, Embra models the processors of a MIPS R3000/R4000 machine faithfully enough to run a commercial operating system and arbitrary user applications. To achieve high simulation speed, Embra uses dynamic binary translation to generate code sequences which simulate the workload. It is the first machine simu ...

20 The impact of architectural trends on operating system performance

M. Rosenblum, E. Bugnion, S. A. Herrod, E. Witchel, A. Gupta

December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles**, Volume 29 Issue 5

Full text available: pdf(2.03 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Results for "(((stalled <and> routine) <and> (code <and> modify))<in>metadata)" e-mail
Your search matched **0** of **1193303** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» View Session History

» New Search

Modify Search

(((stalled <and> routine) <and> (code <and> modify))<in>metadata)

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance revisir

Results for **"((stalled application)<in>metadata)"**

e-mail

Your search matched **0** of **1193303** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» View Session History

» New Search

Modify Search

» Key

((stalled application)<in>metadata)

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Check to search only within this results set

Display Format: Citation Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance revisir